REMARKS

The Examiner's Action mailed on January 17, 2008, has been received and its contents carefully considered. Additionally attached to this Amendment is a Petition for a One-month Extension of Time, extending the period for response to May 17, 2008.

Claims 1 and 9 are the independent claims, and claims 1-14 remain pending in the application. For at least the following reasons, it is submitted that this application is in condition for allowance.

Claims 1-8 were rejected under 35 U.S.C. §103(a) as being obvious over *Lu et al.* (U.S. 7,139,267 B2) in view of *Juszkiewicz et al.* (U.S. 2003/0172797 A1). This rejection is respectfully traversed.

Claim 1 recites:

> A multiple port single chip Ethernet switch comprising at least the following component parts:
> a physical layer entity (PHY) including a plurality of ports;
> an address table for being written to and read out information to operate the plurality of ports;
> a switch for switching the Ethernet switch to a daisy chain test mode; and
> an address resolution control logic including a source address learning engine for performing a packet source address learning process under the daisy chain test mode to deliver a test packet through the plurality of ports progressively from a start transmission port to a stop receiving port to test the chip;
> wherein said component parts of said Ethernet switch are formed on said single chip.

In the present application, the "physical layer entity" (PHY), the "address table", the "switch", and the "address resolution control logic" are included in a

single Ethernet switch. In other words, this application relates to a single Ethernet switch and its test method rather than a network switch system with multiple Ethernet switches and its operating procedures, and hence claim 1 enumerates each of the quoted component parts of the Ethernet switch and then recites "wherein said component parts of said Ethernet switch are formed on said single chip".

In contrast, *Lu et al.* discloses a network switch system with a ring structure or a two-level structure. In the ring structure, four Ethernet switches, ESW 9, ESW 10, ESW 11, and ESW 12 connect sequentially and circularly, and this structure consumes additional processing resources and limits the throughput of the switching system. See FIG. 3 and col. 3, lines 7-12 thereof:

> The process consumes additional processing resources of the switches and limits the throughput of the switching system. Allayer Communications and PMC-Sierra, Inc., both of the Unites States, provide Ethernet switch products that employ similar ring-stacking structures.

In the two-level structure, four lower-level Ethernet switches, ESW17, ESW18, ESW19, and ESW20, are connected to an upper-level Ethernet switch ESW21, and Ethernet switches from different manufacturers or of different designs can be combined to provide a switch system with an increased number of ports. See FIG. 5 and col. 3, lines 35-50 of *Lu et al.*:

> Other than employing switching various fabrics, Ethernet switches may be connected through Ethernet ports in a two-level structure to form a switch system

> that provides more ports. FIG. 5 illustrates a functional block diagram of a two-level structure where four lower-level Ethernet switches, **ESW17**, **ESW18**, **ESW19**, and **ESW20**, are connected to an upper-level Ethernet switch **ESW21**. Whenever any of the lower level Ethernet switches **ESW17**, **ESW18**, **ESW19**, or **ESW20** needs to send a packet, the lower level Ethernet switch first sends the packet to the upper level Ethernet switch **ESW21**, and the packet is then sent to the intended destination through **ESW21**. The two-level structure system requires no specific modification or re-design of Ethernet switches. As a result, Ethernet from different manufacturers or of different designs can be combined to provide a switch system with an increased number of ports.

The two-level structure is frequently employed in systems because of its simplicity, however, the problem of packet loss comes from the limited bandwidth of the connections between the upper-level Ethernet switch ESW21 and the lower-level Ethernet switches ESW17, ESW18, ESW19, and ESW20. See FIG. 5 and col. 3, lines 57-63 thereof:

> The problem of packet loss comes from the limited bandwidth of the connections between the upper-level Ethernet switch **ESW21** and the lower-level Ethernet switches **ESW17**, **ESW18**, **ESW19**, and **ESW20**. Nevertheless, because of its simplicity, this structure is frequently employed in systems that emphasize less on non-blocking features or traffic handling capacities.

In *Lu et al.*, for an Ethernet switch system to provide more ports, the system frequently has to combine, or "stack," two or more Ethernet switches and provides interconnections between the Ethernet switches to enable coordinated operation of the combined system. See col. 1, lines 55-63:

> Each Ethernet switch has only a limited number of ports and manages network traffic among a limited number of nodes or segments. For an Ethernet switch system to provide more ports, the system frequently has to combine, or "stack," two or more Ethernet switches and provides interconnections between the Ethernet switches to enable coordinated operation of the combined system.

> stacking can categorized into one of four types: shared bus, switching fabric, ring-bus, and interleaved interconnection structures.

That is, *Lu et al.* relates to a network switch system with multiple Ethernet switches rather than a single Ethernet switch, and the technique of *Lu et al.* is different from that of this application.

Further, in *Lu et al.*, the switch system includes two or more Ethernet switches to increase number of ports, and *Lu et al.* fails to teach or suggest that the physical layer entity, the address table, the switch, and the address resolution control logic are included in a single Ethernet switch, as recited in claim 1.

Moreover, col. 3, lines 7-12 of *Lu et al.* discloses the manufacturers and the drawbacks of the ring structure, and fails to teach or suggest a physical layer entity including a plurality of ports.

Col. 3, lines 35-50 and 57-65 of *Lu et al.*, as also quoted above, disclose operating procedures, drawbacks and advantages of the two-level structure, and fail to teach or suggest "an address table for being written to and read out information to operate the plurality of ports" as recited in claim 1.

Furthermore, *Lu et al.* fails to teach or suggest "a switch for switching the Ethernet switch to a daisy chain test mode" or "an address resolution control logic including a source address learning engine for performing a packet source address learning process under the daisy chain test mode to deliver a test packet

through the plurality of ports progressively from a start transmission port to a stop receiving port to test the chip" as recited in claim 1.

Lu et al. Therefore fails to teach all the features recited in claim 1.

The Office Action admits that *Lu et al.* fails to disclose "the test source address learning engine for performing a packet source learning process under the daisy chain test mode to deliver a test packet through the plurality of ports progressively from a start transmission port to a stop receiving port to test the chip, wherein said component parts of said Ethernet switch are formed on said single chip", and alleges that this is taught in FIG. 11 and ¶[0149] of *Juszkiewicz et al.*, which reads as follows:

> [0149] The process above should be followed for each device in the system except for the last device. The Nth device in the system, which represents the other end point in the daisy chain should address itself with the number provided in the incoming message and then send an "Address offset return" message back to the address provided in the source address field (usually the STM). The "Address offset return" message should use the "base address"(STM) as a destination address, and the device's own address as the source address. The data field should equal the device address plus one.

Juszkiewicz et al. discloses a system and method that facilitates the interconnection of one or more diverse musical instruments and related audio components on a universal network for purposes of communication of audio signals and signals to identify and control the devices, and does not disclose a single Ethernet switch and its test method. In *Juszkiewicz et al.*, the STM knows

how many devices there are on the network by issuing an Enumerate Device message. See ¶¶[0140] and [0141]:

> Initial Network Enumeration
>
> [0140] After powering up, the STM initializes itself as address 0x0000 and issues an Enumerate Device message on all its connected ports with Control Data set to the next address: 1. The next device receives that packet, assigns itself the address 1, and retransmits the packet to the next device in the daisy chain with Control Data set to the next address: 2. The process continues until all devices are enumerated.
>
> [0141] When an end-point is reached, that device must issue an Address Offset Return message back to the STM with Control Data set to the next address in order to notify it of the number of devices on the network. Upon processing the Address Offset Return message, the STM can be sure that the network is enumerated and it also knows how many devices there are on the network

That is, the Enumerate Device message is delivered between the different devices to determine the number of devices in the network.

However, in the present application, the test packet is delivered through a plurality of ports, and the plurality of ports are located on a single Ethernet switch. Specifically, the test packet is delivered in a single Ethernet switch to test the single Ethernet switch.

Juszkiewicz et al. relates to universal digital media communications and a control system and method, and fails to teach or suggest that a single Ethernet switch includes a physical layer entity, an address table, a switch, and an address resolution control logic, as recited in claim 1.

¶[0149] of Juszkiewicz et al., as quoted above, discloses delivering procedures for the message between the STM and the last (Nth) device in the

system, and fails to teach or suggest "an address resolution control logic including a source address learning engine for performing a packet source address learning process under the daisy chain test mode to deliver a test packet through the plurality of ports progressively from a start transmission port to a stop receiving port to test the chip" as recited in claim 1.

Moreover, the drawings in *Juszkiewicz et al.* are numbered from FIG. 1 to FIG. 10b, and there is no FIG. 11 in *Juszkiewicz et al.*, so it is not clear what view is referred to in the Office Action.

Juszkiewicz et al. therefore also fails to teach or suggest all the features of claim 1.

Consequently, neither *Lu et al.* nor *Juszkiewicz et al.*, whether taken separately or in combination, teaches or suggests all the features recited in claim 1, and claim 1 is allowable, along with claims 1-8 that depend therefrom.

Claims 9-14 were rejected under 35 U.S.C. §103(a) as being obvious over *Duvvury* (U.S. 2005/0213560 A1) in view of *Lu et al.* This rejection is respectfully traversed.

Claim 9 recites:

> A daisy chain test for a single chip Ethernet switch integrated with a physical layer entity including a plurality of ports, the switch having an address table for being written to and read out information to operate the plurality of ports, the test comprising the steps of:
> connecting each of the plurality of ports to a respective passive loop-back device;
> selecting a start transmission port and a stop receiving port from the plurality of ports;

supplying a test packet to the start transmission port; and
proceeding a packet source address learning process for delivering the test packet from the start transmission port to the stop receiving port progressively, wherein the step of proceeding employs a source address learning engine with a daisy chain testing function; and
determining a test result by verifying a last received packet at the stop receiving port.

Duvvury discloses a method of creating a cluster of Ethernet switches depending on each particular network configuration. When the switches are connected in a daisy-chain topology, the candidate switch that is connected to the commander switch is added first, and then each subsequent switch in the chain is added as it is discovered by CDP. See FIG. 9 and ¶[0075]:

> [0075] The method of creating a cluster of Ethernet switches depends on each particular network configuration. If the switches are arranged in a star topology, as in FIG. 8, with the commander switch at the center, all of the member switches may be added to the cluster at once. On the other hand, if the switches are connected in a daisy-chain topology, as in FIG. 9, the candidate switch that is connected to the commander switch is added first, and then each subsequent switch in the chain is added as it is discovered by CDP. If switches are daisy-chained off a star topology, as in the exemplary hybrid configuration shown in FIG. 10, all the switches that are directly connected to the commander switch may be added first, and then the daisy-chained switches may be added one at a time.

Duvvury fails to teach or suggest a daisy chain test for a single chip Ethernet switch, as per the present invention.

In ¶[0024] of *Duvvury*, an Ethernet LAN switch improves bandwidth by separating collision domains and selectively forwarding traffic to the appropriate segments:

> [0024] An Ethernet LAN switch improves bandwidth by separating collision domains and selectively forwarding traffic to the appropriate segments. FIG. 3 illustrates the topology of a typical Ethernet network **40** in which a LAN switch **42** has been

> installed. With reference now to FIG. 3, exemplary Ethernet network **40** includes a LAN switch **42**. As shown in FIG. 3, LAN switch **42** has five ports: **44**, **46**, **48**, **50**, and **52**. The first port **44** is connected to LAN segment **54**. The second port **46** is connected to LAN segment **56**. The third port **48** is connected to LAN segment **58**. The fourth port **50** is connected to LAN segment **60**. The fifth port **52** is connected to LAN segment **62**. The Ethernet network **40** also includes a plurality of servers **64-A-64-C** and a plurality of clients **66-A-66-K**, each of which is attached to one of the LAN segments **54**, **56**,**58**, **60**, or **62**. If server **64-A** on port **44** needs to transmit to client **66-D** on port **46**, the LAN switch **42** forwards Ethernet frames from port **44** to port **46**, thus sparing ports **48**, **50**, and **52** from frames destined for client **66-D**. If server **64-C** needs to send data to client **66-J** at the same time that server **64-A** sends data to client **66-D**, it can do so because the LAN switch can forward frames from port **48** to port **50** at the same time it is forwarding frames from port **44** to port **46**. If server **64-A** on port **44** needs to send data to client **66-C**, which is also connected to port **44**, the LAN switch **42** does not need to forward any frames.

Further, FIG. 3 shows an exemplary Ethernet network **40** including a LAN switch **42**. Specifically, *Duvvury* discloses the topology of the Ethernet network **40** including the LAN switch **42**, a plurality of LAN segments **54-62**, a plurality of servers **64-A-64-C** and a plurality of clients **66-A-66K**.

Duvvury fails to teach or suggest a "single chip Ethernet switch integrated with a physical layer entity including a plurality of ports" or "the switch having an address table for being written to and read out information to operate the plurality of ports" as recited in claim 9.

Further, see FIG. 4 and ¶[0027] of *Duvvury*:

> [0027] Referring now to FIG. 4, two LAN switches **70-A** and **70-B** are shown, connected in a cascaded configuration. As shown, each of the LAN switches **70-A** and **70-B** contains eight ports, **72-A-72-H** and **74-A-74-H**. On each of the LAN switches **70-A** and **70-B**, four ports **72-A-72-D** and **74-A-74-D** are connected to computer workstations **76-A-76-D** and **76-E-76-H**, respectively. The other four ports on each LAN switch (i.e., ports **72-E-72-H** on LAN switch **70-A**, and ports **74-E-74-H** on LAN switch **70-B**) are dedicated to interswitch communication. For example, if each of the four interswitch connections is capable of supporting a 100 Mbps Fast Ethernet channel, the aggregate interswitch communication rate of the

> switches connected as shown in FIG. 4 is 400 Mbps. However, the total number of ports available for connecting to workstations or other network devices on each LAN switch is diminished due to the dedicated interswitch connections that are necessary to implement the cascaded configuration.

Each of the LAN switches **70-A** and **70-B** contains eight ports, **72-A-72-H** and **74-A-74-H**. On each of the LAN switches **70-A** and **70-B**, four ports **72-A-72-D** and **74-A-74-D** are connected to computer workstations **76-A-76-D** and **76-E-76-H**, respectively. The other four ports on each LAN switch (i.e., ports **72-E-72-H** on LAN switch **70-A**, and ports **74-E-74-H** on LAN switch **70-B**) are dedicated to interswitch communication. That is, *Duvvury* discloses that the two LAN switches **70-A** and **70-B** are connected in a cascaded configuration, and fails to teach or suggest "connecting each of the plurality of ports to a respective passive loop-back device" and "selecting a start transmission port and a stop receiving port from the plurality of ports" as recited in claim 9.

In FIG. 2B and ¶[0023] of *Duvvury*, the discovery protocol logic **230** receives, processes, and sends Cisco Discovery Protocol ("CDP") or other discovery protocol packets to neighboring network devices on the network:

> [0023] According to embodiments of the present invention, discovery protocol logic **230** receives, processes, and sends Cisco Discovery Protocol ("CDP") or other discovery protocol packets to neighboring network devices on the network. Packet redirection logic **260** examines the source and destination addresses of Ethernet packets under control of the configuration and management interface **270** and forwards them to other network devices in a cluster configuration. As known to those skilled in the art, the program code corresponding to discovery protocol logic **230**, learning logic **240**, forwarding logic **250**, packet redirection logic **260**, configuration and management interface **270**, and other necessary functions may all be stored on a computer-readable medium. Depending on each particular application, computer-readable media suitable for this purpose may include,

without limitation, floppy diskettes, hard drives, RAM, ROM, EEPROM, nonvolatile RAM, or flash memory.

The packet redirection logic **260** examines the source and destination addresses of Ethernet packets under control of the configuration and management interface **270** and forwards them to other network devices in a cluster configuration. The program code corresponding to discovery protocol logic **230**, learning logic **240**, forwarding logic **250**, packet redirection logic **260**, configuration and management interface **270**, and other necessary functions may all be stored on a computer-readable medium. That is, *Duvvury* discloses the operating procedures that the packet delivers between the network devices on the network, and a packet destination address is determined based on the command from the management station.

Duvvury fails to teach or suggest "supplying a test packet to the start transmission port" and "proceeding a packet source address learning process for delivering the test packet from the start transmission port to the stop receiving port progressively" as recited in claim 9.

In FIG. 9 and ¶[0067] of *Duvvury*, in cluster **110**, only member switch **102-A** is directly connected to the commander switch **100**, and member switches **102-B**-**102-G** are each connected to an "upstream" switch and to a "downstream" switch:

[0067] A second example of a cluster configuration, known as a "daisy chain" configuration, is shown in FIG. 9. In cluster **110**, only member switch **102-A** is directly connected to the commander switch **100**. Member switches **102-B**-**102-G**

> are each connected to an "upstream" switch (one that is fewer "hops" away from commander switch **100**) and to a "downstream" switch (one that is more "hops" away from commander switch **100**). Finally, the last switch in the chain (member switch **102-H**) is only connected to its upstream "neighbor" **102-G**.

That is, *Duvvury* discloses a cluster configuration with "daisy chain" configuration rather than a daisy chain testing function. *Duvvury* fails to teach or suggest "wherein the step of proceeding employs a source address learning engine with a daisy chain testing function" and "determining a test result by verifying a last received packet at the stop receiving port" as recited in claim 9.

The Office Action admits that *Duvvury* does not disclose an "a single chip Ethernet switch", and alleges that this is taught in ¶[0149] of *Lu et al.* However, there is no ¶[0149] in *Lu et al.*, which also fails to teach or suggest "a single chip Ethernet switch" as recited in claim 9.

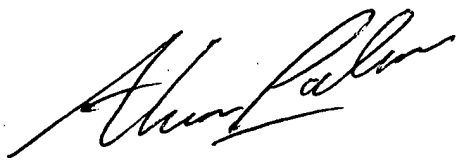
Consequently, neither *Duvvury* nor *Lu et al.*, whether taken separately or in combination, teaches or suggests all the features recited in claim 9, and therefore claim 9 is allowable, together with claims 10-14 that depend therefrom.

It is submitted that this application is in condition for allowance. Such action and the passing of this case to issue are requested.

Should the Examiner feel that a conference would help to expedite the prosecution of this application, the Examiner is hereby invited to contact the undersigned counsel to arrange for such an interview.

Should the remittance be accidentally missing or insufficient, the Commissioner is hereby authorized to charge the fee to our Deposit Account No. 18-0002, and advise us accordingly.

Respectfully submitted,



May 13 ,2008
Date

Alun L. Palmer – Reg. No. 47,838
RABIN & BERDO, PC – Cust. No. 23995
Facsimile: 202-408-0924
Telephone: 202-371-8976

ALP/pq